# Gauge Equivariant Deep Q-Learning on Discrete Manifolds

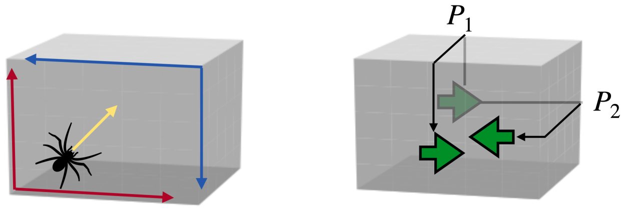
**Sourya Basu** [*], **Pulkit Katdare** [*], **Katherine Driggs-Campbell, Lav R. Varshney**
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
{sourya, katdare2, krdc, varshney}@illinois.edu

## Abstract

Data, at any point on a manifold, can be represented on the tangent plane at that point with respect to a basis, called a gauge. But the choice of gauge is not unique for arbitrary manifolds. Hence, for agents traversing an environment embedded on a manifold, the same environment may appear differently if the choice of gauge changes or when moving to a different point that has a different gauge. This may be deleterious to an agent's learning, as compared to learning on, say, a flat grid where it is easy to choose a fixed gauge for each point. To this end, we provide a formulation of deep Q-learning that learns policies (and Q-values) that are equivariant (invariant) to changes in choice of gauge. This leads to an efficient learning algorithm independent of the choice of gauge. Our experimental results demonstrate significant improvement in learning on novel environments embedded in arbitrary manifolds such as spheres, hills, and urns, compared to naive approaches.

## 1 Introduction

Consider the problem of deep Q-learning on discrete manifolds. Data can be represented on the tangent plane of any manifold using a basis, called a *gauge*. But for general manifolds, the choice of gauge is not unique. Hence, the same *geometric* data may appear differently based on the choice of the gauge, as shown in Fig. 1a, where a spider (agent) crawls on the surface of a cube with each surface having a different choice of gauge. The observations made by the spider change with the choice of gauge. Moreover, Fig. 1b shows the challenge of *parallel transporting* (De Haan et al., 2020) data along different paths between the same points, resulting in different features being transported. Hence, an agent moving between two points along different paths on a manifold may observe the data differently. These two challenges do not arise on flat surfaces, where gauges can be fixed easily and moving parallelly along different paths to the same point yields the same observation. We encounter both these challenges by first arbitrarily fixing gauges at each point and then providing an efficient gauge equivariant framework of deep Q-learning.



(a) Choice of gauge

(b) Parallel transport

Figure 1: Choice of gauge in (a) results in changed observations of the same data and (b) parallel transporting on a manifold along different paths may result in changed observation of the same data.

[*] Equal contribution

## 2 Preliminaries

**Equivalence and equivariance:** Two points $x, x' \in \mathcal{X}$ are called *$f$-equivalent* for some function $f : \mathcal{X} \mapsto \mathcal{Y}$ if $f(x) = f(x')$. A function $f : \mathcal{X} \mapsto \mathcal{Y}$ is *equivariant* to sets of transformations $\{L_g : \mathcal{X} \mapsto \mathcal{X} | g \in G\}$ and $\{K_g : \mathcal{Y} \mapsto \mathcal{Y} | g \in G\}$ if $K_g(f(x)) = f(L_g(x))$ for all $g \in G$, for some group $G$. The function $f : \mathcal{X} \mapsto \mathcal{Y}$ is called *invariant* to the set of transformations $\{L_g : \mathcal{X} \mapsto \mathcal{Y} | g \in G\}$ if $K_g$ is the identity function, i.e. $f(x) = f(L_g(x))$ for all $g \in G$.

**Gauge and gauge transformations** We define a discrete mesh $M_D$ by a set of vertices in $\mathbb{R}^3$, with a set of faces $\mathcal{F}$ of tuples of vertices describing their corners. Further, for $M_D$ to be a discrete 2D manifold, we require each edge to be connected to two faces and the neighborhood of each vertex to be homeomorphic to a disk. At each node $v \in \mathcal{V}$, we define a vertex normal $N_v$, as the weighted sum of the normals of the faces that include $v$. We define the plane perpendicular to $N_v$ as the tangent plane, $T_v M_D$, at $v$. In this discrete case, defining the gauge at any vertex $v$ simplifies to choosing a reference neighbor vertex, say, $v_q$. The frames on the tangent plane becomes the line joining $v$ to the projection of $v_q$ on the tangent plane at $v$, and the line perpendicular to it. Features at any point $v$ on $M_D$ are represented on the tangent plane, $T_v M_D$. Gauge on $M_D$ is defined as a position-dependent invertible linear map $w_v : \mathbb{R}^2 \mapsto T_v M_D$. Thus, if $\{e_1, e_2\}$ is the standard basis of $\mathbb{R}^2$, then $\{w_v(e_1), w_v(e_2)\}$ defines the basis of $T_v M_D$.

Gauge transformations, $g_v$, are point-dependent transformations of gauges on the manifold, which are described as $d \times d$ invertible matrices, $\mathbf{GL}(d, \mathbb{R})$. Depending on the transformation of matrices we consider, we can restrict the allowed $g_v$ matrices, also called reduction of the structure group $\mathbf{GL}(d, \mathbb{R})$. In our case, we consider orientable manifolds, for which we use $g_v \in \mathrm{SO}(2)$, i.e. the set of orthogonal matrices with positive determinant in $\mathbf{GL}(2, \mathbb{R})$.

**MDP and MDP homomorphism** A Markov Decision Process (MDP), $M$, is given by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T}$ gives the transition probabilities $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}_{\geq 0}$, and $\gamma \in [0, 1]$ is the discount factor. The goal of an MDP is to find a policy $\pi \in \Pi$, $\pi : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}_{\geq 0}$, such that $\sum_{a \in \mathcal{A}} \pi(a|s) = 1$ for all $s \in \mathcal{S}$, which maximizes the expected reward $\mathrm{R}_t = \mathbb{E}_\pi[\sum_{k=0}^{T} \gamma^k r_{t+k+1}]$. Two important associated terms with a policy $\pi$ are its value function $V^\pi(s)$ and $Q^\pi(s, a)$, where $V^\pi(s)$ is the expected reward at state $s$, and $Q^\pi(s, a)$ is the expected reward on playing $a$ at state $s$ under policy $\pi$. $V^\pi(s)$ and $Q^\pi(s, a)$ are related by the Bellman equations (Bellman, 1957) and optimal policies $\pi^*$ result in optimal $V^*(s)$ and $Q^*(s, a)$.

An MDP homomorphism, $h$, consists of a tuple of surjective maps $(\sigma, \{\alpha_s | s \in \mathcal{S}\})$ from the state-action space $\mathcal{S} \times \mathcal{A}$ of the MDP $M$ to the abstract state-action space $\bar{\mathcal{S}} \times \bar{\mathcal{A}}$ of an MDP $\bar{M} = (\bar{\mathcal{S}}, \bar{\mathcal{A}}, \bar{\mathcal{R}}, \bar{\mathcal{T}}, \bar{\gamma})$, where $\sigma : \mathcal{S} \mapsto \bar{\mathcal{S}}$ and $\alpha_s : \mathcal{A} \mapsto \bar{\mathcal{A}}$. Further, these maps must satisfy the following conditions:

$$\bar{\mathcal{R}}(\sigma(s), \alpha_s(a)) = \mathcal{R}(s, a) \qquad \text{for all } s \in \mathcal{S}, a \in \mathcal{A} \tag{1}$$

$$\bar{\mathcal{T}}(\sigma(s') | \sigma(s), \alpha_s(a)) = \sum_{s'' \in \sigma - 1(s')} \mathcal{T}(s'' | s, a) \qquad \text{for all } s \in \mathcal{S}, a \in \mathcal{A}. \tag{2}$$

## 3 Method

Our main insight is that the outcome of a policy for an MDP with state-action space defined on a manifold should be independent of the gauge of its representation. That is, the policy is an intrinsic property of the environment, irrespective of the choice of gauge. Consider the example in Fig. 1a, where a spider moves on the faces of a cube. The same direction of motion with respect to the ambient space, $\mathbb{R}^3$, appears differently with different choices of gauges, but, intrinsically it is the same environment and the policy learned should not be independent of the choice of gauge. Hence, we introduce policy networks (and Q-networks) that are equivariant (invariant) to the change of gauge on a manifold.

**Gauge-equivariant MDP homomorphism** First we look at some equivalence relations for MDP on manifolds.

$$\mathcal{R}(s,a) = \mathcal{R}(\rho_g(s), \rho_g(a)) \qquad \text{for all } s \in \mathcal{S}, a \in \mathcal{A} \tag{3}$$

$$\mathcal{T}(s'|s,a) = \mathcal{T}(\rho_g(s')|\rho_g(s), \rho_g(a)) \qquad \text{for all } s \in \mathcal{S}, a \in \mathcal{A}, \tag{4}$$

where $\rho_g \in \mathbf{GL}(2,\mathbb{R})$ is a representation of $g \in G$, denoting a transformation of gauge. Note that the equivalence relations obtained are similar to the ones obtained by van der Pol et al. (2020), but, we do not assume any symmetry in our environment as in (van der Pol et al., 2020). Symmetries are often hard to find in any system let alone an RL problem. Our relation comes from the insight that change in representation of the system does not change its dynamics.

We define gauge equivariant MDPs by defining $h : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{O}(\mathcal{S} \times \mathcal{A})$ that maps each state-action pair to their respective orbits under a set of gauge transformations, say, $G$. From Ravindran & Barto (2001), we know that $h$-equivalent state-action pairs share the same optimal $Q$ and $V$ functions. Moreover, there exist corresponding abstract $\bar{Q}$ and $\bar{V}$ functions that obtain these optimal values in the abstract MDP, $\bar{M}$, obtained from the map $h$. Further, policies learnt in $\bar{M}$, can be *lifted* to $M$ as shown in equation 5, which is optimal in $M$.

$$\pi^{\uparrow}(a|s) := \frac{\bar{\pi}(\bar{a}|\sigma(s))}{|\{a \in \alpha_s^{-1}(\bar{a})\}|} \qquad \text{for any } s \in \mathcal{S}, a \in \mathcal{A}, \tag{5}$$

where $\bar{a} = \alpha_s(a)$. Using the above definition of lifting, it is easy to see that $\pi^{\uparrow}(a|s) = \pi^{\uparrow}(a'|s')$, where $s' = \rho_g(s), a' = \rho_g^{s'}(a)$ for $g \in G$. Thus, similar to van der Pol et al. (2020), we have the relation $\boldsymbol{\pi}(\rho_g(s)) = \rho_g(\boldsymbol{\pi}(s))$. But, in our framework of meshes, the action space may be irregular, unlike in (van der Pol et al., 2020), where the problems considered mostly have fixed action space. E.g., in a gridworld-like problem where the agent moves on a mesh to find a goal by traversing on the mesh, at every node, the possible neighbors are different and at different angles with respect to each other. The possible orientations of the neighbors at each node are continuous and infinite. Hence, instead of constructing equivariant policy networks like in (van der Pol et al., 2020), we construct gauge-invariant Q-networks that output the Q-values for any pair of state-action. For Q-values, the equivariance relation on policy is the same as invariance relation on the Q-function, i.e. $Q(s,a) = Q(\rho_g(s), \rho_g(a))$ for $g \in G$. Thus, we need $Q$ to be invariant to possible change in gauges on a manifold, e.g. for environments lying on the surface of a cube we would need equivariance to $90°$ rotations, whereas for general manifolds with state-actions lying on the tangent plane we would need $SO(2)$-equivariance, i.e. equivariance to transformations by arbitrary 2D angles.

## 4 Related Works

Group equivariant networks are very well studied (Cohen & Welling, 2016; Cohen et al., 2018; Ravanbakhsh et al., 2017). In RL, group equivariant MDP was proposed exploiting symmetries in RL environments (van der Pol et al., 2020). But, searching for symmetries in data is a non-trivial problem and is an active area of research (Zhou et al., 2020; Dehmamy et al., 2021; Basu et al., 2021; Finzi et al., 2021). In contrast, we do not assume any symmetry in the environment. We focus on learning independent of representation of the environment based on gauges by developing a gauge equivariant deep Q-learning framework.

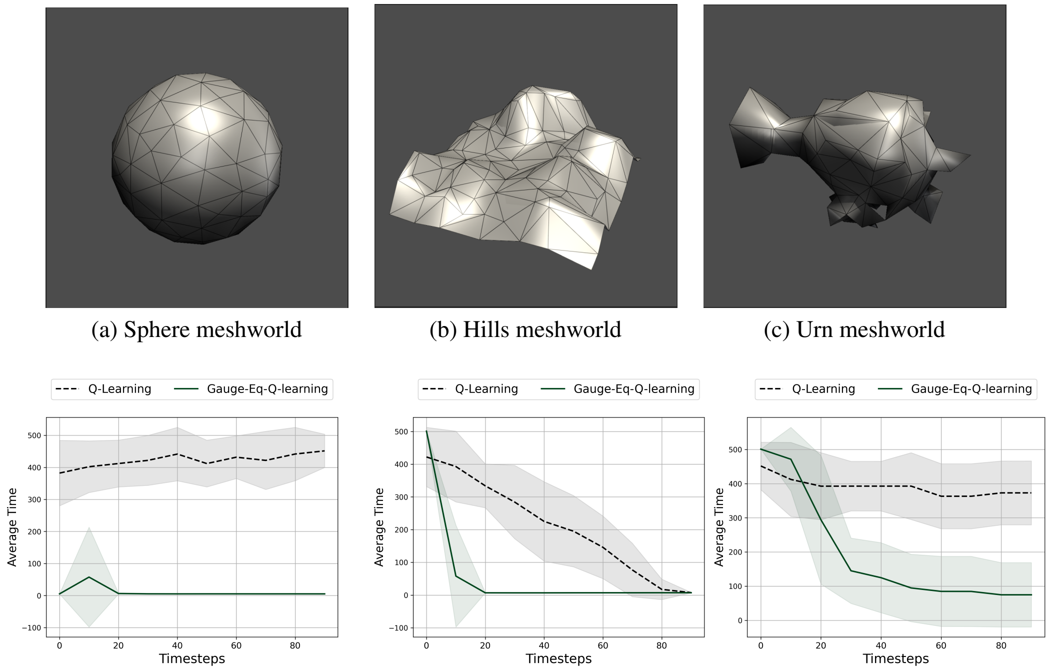
## 5 Experiments

We first describe the environments followed by the results. Details of the equivariant network construction method and hyperparameters used are given in Appendix A.

**RL Environments on a manifold** We consider the basic problem of gridworld embedded on different manifolds like a sphere, hills, or an urn. We call this environment *meshworld*. Here, an agent starting at some random point on a mesh wants to find the goal, which is another randomly picked point on the mesh. The observation of the agent is the vector starting from its location pointing to the goal, but projected on the tangent plane corresponding to the location of the agent. The gauge at each location is chosen arbitrarily but fixed for training. For evaluation, similarly, we choose arbitrarily the gauges at each location, but they are not necessarily the same as in training. This is because evaluation maybe performed separately from training and there is no unique choice of gauges. The meshes chosen are shown in Fig. 2, which were all created using the PyVista software (Sullivan & Kaszynski, 2019) ensuring that each one has 100 nodes.

The action space at any node is the set of neighbors and the reward $r(s, a)$ for any state $s$ and action $a$ is inversely proportional to the geodesic distance of the next state to the goal. For updating states, we also need to take care of gauge change and avoid the problem of parallel transport illustrated in Fig. 1b. Going from a state $s_1$ at node $p$ to a state $s_2$ at a neighbor $q$, we need to ensure that we also take care of the change in gauge from node $p$ to $q$. This can either be done by projecting the state on the new gauge after updating the 3D state vector or by changing the magnitude of the current state vector appropriately and multiplying by a change of gauge matrix, $\rho(p \to q)$.

**Results** The results of our experiments on meshworld in Fig. 2 (d), (e), (f) show huge gains in average time taken by the agent to reach the goal in each of the manifolds considered: sphere, hills, and urn. The results indicate the advantage of using gauge equivariance in our deep Q-learning formulation. All plots shown are results averaged over 10 runs with fixed seeds. Moreover, the gains obtained from equivariance can be seen very early in the training, indicating better sample efficiency of our method. We also conduct more experiments on meshes of different sizes showing similar gains, as illustrated in Appendix B.



(a) Sphere meshworld

(b) Hills meshworld

(c) Urn meshworld

(d) Sphere results

(e) Hills results

(f) Urn results

Figure 2: Environments on meshes used for our experiments, each having 100 nodes are shown in (a), (b), (c). Average steps taken in each mesh environment as a function of number of training episodes are shown in (d), (e), (f). Each plot is averaged over 10 runs over fixed seeds.

## 6 Conclusion

We propose a natural framework of deep Q-learning on meshes by addressing the problem of choice of gauge and parallel transport on meshes. We release a novel RL environment called *meshworld*, where the environment is embedded on meshes. We show that our framework of gauge equivariant deep Q-learning addresses the above-mentioned challenges and experimental evidence confirms huge gains over traditional deep Q-learning frameworks.

## Acknowledgement

This work was supported in part by the Department of Energy (DOE) award DE-SC0012704.

## References

Sourya Basu, Akshayaa Magesh, Harshit Yadav, and Lav R Varshney. Autoequivariant network search via group decomposition. *ArXiv:2104.04848*, 2021.

Richard Bellman. *Dynamic Programming*. Princeton Univ. Press, Princeton, NJ, USA, 1957.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning*, 2016.

Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations*, 2018.

Pim De Haan, Maurice Weiler, Taco Cohen, and Max Welling. Gauge equivariant mesh CNNs: Anisotropic convolutions on geometric graphs. In *International Conference on Learning Representations*, 2020.

Nima Dehmamy, Robin Walters, Yanchen Liu, Dashun Wang, and Rose Yu. Automatic symmetry discovery with lie algebra convolutional network. *Advances in Neural Information Processing Systems*, 2021.

Marc Finzi, Max Welling, and Andrew Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In *International Conference on Machine Learning*, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Siamak Ravanbakhsh, Jeff Schneider, and Barnabás Póczos. Equivariance through parameter-sharing. In *International Conference on Machine Learning*, 2017.

Balaraman Ravindran and Andrew G Barto. Symmetries and model minimization in Markov decision processes, 2001.

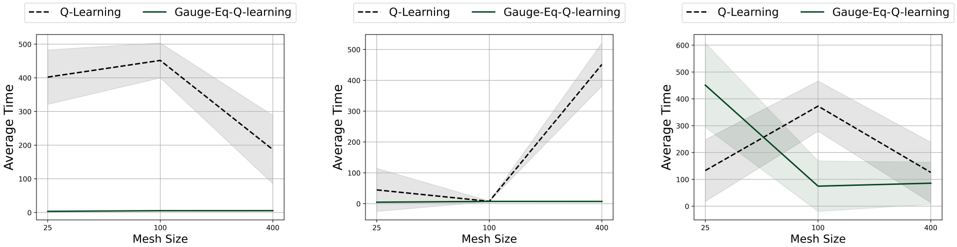
C Sullivan and Alexander Kaszynski. PyVista: 3D plotting and mesh analysis through a streamlined interface for the visualization toolkit (VTK). *Journal of Open Source Software*, 4(37):1450, 2019.

Elise van der Pol, Daniel Worrall, Herke van Hoof, Frans Oliehoek, and Max Welling. MDP homomorphic networks: Group symmetries in reinforcement learning. *Advances in Neural Information Processing Systems*, 2020.

Allan Zhou, Tom Knowles, and Chelsea Finn. Meta-learning symmetries by reparameterization. In *International Conference on Learning Representations*, 2020.

## A Equivariant Network Construction Details

For constructing gauge equivariant networks, we use SO(2)-equivariant kernels from (De Haan et al., 2020). The input features to our network is of type $2 \times \rho_1$ with both state and the action represented as $\rho_1$ features. We use two layers of SO(2) equivariant kernels with intermediate layers with 3 channels of representation type $(\rho_0 + \rho_1 + \rho_2)$. The output has 6 channels of representation type $(\rho_0 + \rho_1)$. We take outputs of the norms of each of the six channels and pass it through a fully connected layer with a single output, hence, making the output invariant to change in gauges. For comparison with non-equivariant models, we use fully connected networks with two layers and nearly equal number of parameters. For all cases, we use the Adam optimizer (Kingma & Ba, 2015) with a learning rate of $1e$-4 and weight decay of $1e$-2. For Q-learning, we use a discount factor of $0.99$ and for training we use $\epsilon$-greedy strategy, where epsilon is set to $1$ in the beginning and is decreased exponentially to $1e$-2 using an exponential decay of $1e$-3.



(a) Sphere meshworld

(b) Hills meshworld

(c) Urn meshworld

Figure 3: Average steps taken in each mesh environment with varying number of nodes in each mesh shown in (a), (b), (c). Each plot is averaged over 10 runs over fixed seeds.

## B Experiments with varying size of nodes in meshworld

In this section, we generalize the experiments we conduct in Sec. 5 to varying number of nodes showing that the gain obtained from gauge equivariance shows similar advantages across varying sizes of meshes.